

Get Rich Quick

Time and memory limits: 2 seconds, 256 MB

You've been introduced to an exciting investment opportunity that sounds almost too good to be true. Your new business mentor has prepared N investment packages, which you must purchase in order, from package 1 to package N . **You cannot skip packages or change their order.**

The i th package costs A_i dollars and comes with an incredible guarantee. After paying A_i dollars, you immediately receive a payout of $2A_i$ dollars, giving you a net profit of A_i dollars. You must have enough money to pay for each package before receiving its payout. You then attempt the next package. If you cannot afford it, you stop.

Your starting money must be a non-negative integer number of dollars. Your mentor is considering Q adjustments to the prices of these packages, each affecting exactly one package. For adjustment i , package X_i would instead cost B_i dollars and pay out $2B_i$ dollars. For each proposed adjustment, determine the minimum amount of money you need to start with to purchase every package in the required order.

Each adjustment is considered separately. All other packages keep their original prices and payouts. Adjustments do not carry over between queries.

Of course, your mentor assures you that the scheme is completely sustainable.

Subtasks and constraints

Your program will be graded using many secret tests. Every test follows some rules:

- $1 \leq N, Q \leq 1\,000\,000$.
- $1 \leq A_i \leq 1\,000\,000\,000$ for all i .
- $1 \leq X_i \leq N$ for all i .
- $1 \leq B_i \leq 1\,000\,000\,000$ for all i .

The secret tests are divided into subtasks. Your program must correctly solve **every test** within a subtask to earn the marks for that subtask:

- For Subtask 1 (10 marks), $Q = 1$.
- For Subtask 2 (15 marks), $N \leq 20$.
- For Subtask 3 (20 marks), $X_i = N$ for all i . That is, only the last package's price changes.
- For Subtask 4 (20 marks), $X_i = 1$ for all i . That is, only the first package's price changes.
- For Subtask 5 (35 marks), no special rules apply.

Input

Your solution must read input and print output. We recommend using the solution templates (which you can find on the competition website) to help you with input and output.

The input follows a specific format:

- The 1st line contains the integers N and Q .
- The 2nd line contains $A_1, A_2, \dots, A_N$, the original package prices in dollars.
- The next Q lines describe the adjustments. The i th line contains the integers X_i and B_i .

Output

Your program must output Q lines. The i th line must contain the minimum starting amount of money, in dollars, needed to purchase all the packages in order if only the i th adjustment is made.

Sample input 1

```
4 4
3 8 2 20
2 1
4 1
1 10
2 8
```

Sample output 1

```
14
5
10
7
```

Sample input 2

```
1 3
7
1 1
1 12
1 7
```

Sample output 2

```
1
12
7
```

Explanation

In the 1st sample case:

- For the 1st adjustment, the package prices are 3, 1, 2, 20 dollars. Starting with 14 dollars, you buy the first three packages in order and your money increases to 17, then 18, then 20 dollars. You can then afford the last package. Starting with 13 dollars is not enough: you would reach the last package with only 19 dollars.
- For the 2nd adjustment, the package prices are 3, 8, 2, 1 dollars. Starting with 5 dollars lets you purchase every package in order. With 4 dollars, you would have only 7 dollars after buying the first package and receiving its return, so you could not afford the second package.
- For the 3rd adjustment, the package prices are 10, 8, 2, 20 dollars. You need at least 10 dollars to afford the first package, and this is enough to purchase the remaining packages in order too.
- For the 4th adjustment, the second package keeps its original price of 8 dollars. The prices therefore remain 3, 8, 2, 20 dollars, and the minimum starting amount is 7 dollars.

In the 2nd sample case, there is only one package. You must start with at least enough money to pay its adjusted price, before you receive its return.